Few-Shot Learning for Image Denoising

Bo Jiang, Yao Lu*, Bob Zhang, *Senior Member, IEEE* Guangming Lu*, *Senior Member, IEEE*

***Abstract*—Deep Neural Networks (DNNs) have achieved impressive results on the task of image denoising, but there are two serious problems. First, the denoising ability of DNNs-based image denoising models using traditional training strategies heavily relies on extensive training on clean-noise image pairs. Second, image denoising models based on DNNs usually have large parameters and high computational complexity. To address these issues, this paper proposes a two-stage Few-Shot Learning for Image Denoising (FSLID). Our FSLID is a two-stage denoising strategy integrating Basic Feature Learner (BFL), Denoising Feature Inducer (DFI), and Shared Image Reconstructor (SIR). BFL and SIR are first jointly unsupervised to train on the base image dataset $\mathcal{D}_{base}$ consisting of easily collected high-quality clean images. Following this, the trained BFL extracts the guided features and constraint features for the noisy and corresponding clean images in the novel image dataset $\mathcal{D}_{novel}$, respectively. Furthermore, DFI encodes the noisy features of the noisy images in $\mathcal{D}_{novel}$. Then, inducing both the guided features and noisy features, DFI can generate the denoising prior features for the SIR with frozen weights to adaptively denoise the noisy images. Furthermore, we propose refined, low-channel-count, recursive multi-branch Multi-Scale Feature Recursive (MSFR) to modularly formulate an efficient DFI to capture more diverse contextual features information under a limited number of feature channels. Thus, compared with the baseline models, the FSLID composed of the proposed MSFR can significantly reduce the number of model parameters and computational complexity. Extensive experimental results demonstrate our FSLID significantly outperforms well-established baselines on multiple datasets and settings. We hope that our work will encourage further research to explore the field of few-shot image denoising.**

***Index Terms*—Image denoising, Few-shot learning, Basic feature Learner, Denoising feature induer, Shared image reconstructor, Multi-scale feature recursive.**

I. Introduction

IMAGE denoising is a typical and challenging early visual processing task, which aims to remove noise in images to improve visual quality. Since the image denoising task is mathematically ill-posed, it is an extremely challenging task. Compared with conventional image denoising methods, image denoising methods using Deep Neural Networks (DNNs) [1], [2], [3] have greatly improved in performance. Meanwhile, with the rapid development of deep neural networks, image denoising methods based on DNNs achieve state-of-the-art performance.

*Yao Lu and Guangming Lu are corresponding authors.

Bo Jiang and Yao Lu are with the Department of Computer Science and Technology, Harbin Institute of Technology at Shenzhen, Shenzhen 518057, China (e-mail: jiangbo_PhD@outlook.com; luyao2021@hit.edu.cn).

Guangming Lu is with the Department of Computer Science and Technology, Harbin Institute of Technology at Shenzhen, and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518057, China (e-mail: luguangm@hit.edu.cn)

Bob Zhang is with the PAMI Research Group, Department of Computer and Information Science, University of Macau, Macau, China (e-mail: bobzhang@um.edu.mo)

TABLE I. Abbreviations in the proposed method.

Abbreviations	Full Name	Sections of Description
BFL	Basic Feature Learner	Section IV-A
SIR	Shared Image Reconstructor	Section IV-B
DFI	Denoising Feature Inducer	Section IV-C
MSFR	Multi-Scale Feature Recursive	Section IV-C
FSLID	Few-Shot Learning for Image Denoising	Section IV-D

Even though DNNs have achieved impressive results on the task of image denoising, there are subjected to two serious problems. ***(a) The image denoising ability based on DNNs relies heavily on a large number of clean-noise image pairs.*** Due to the difficult and expensive collection of such training image pairs from real-world scenarios, the application of CNNs will be gravely restricted in real-world scenarios. ***(b) Moreover, image denoising models based on DNNs often have huge parameters and high computational complexity while improving performance.*** Image denoising performance improves significantly with the increase of DNNs capacity (*i.e.*, depth and width), but this leads to excessively large parameters of DNNs-based image denoising methods. At the same time, compared with DNNs for classification tasks, since the image denoising method belongs to image-to-image, the size of the feature map in the latent feature space in DNNs is large, thus this may cause the computational complexity of the methods to soar. Therefore, there is an urgent need to study an efficient image denoising method that only considers a small number of clean-noise image pairs.

Humans, by contrast, are competent at learning new visual understandings from a few examples, generalizing with easiness to new examples [4], [5], [6], [7]. Such ability to learn from a few examples is also desired for computer vision systems, especially in image denoising tasks. Recently, the few-shot classification [8], [9], [10] provides a reference and promising strategy for learning from only a few examples. However, image denoising is by nature much more different from image classification, because it is a pixel-level regression problem. Thus, current few-shot classification methods cannot be directly applied to the few-shot image denoising problem. Considering that few-shot learning can obtain strong generalization ability only through a small number of samples or feature descriptions and the powerful image representation ability of DNNs, the key difficulties and challenges in designing an efficient image denoising method with few-shot learning mainly come from two aspects.

(1) How to use the few-shot learning to obtain prior feature knowledge from only a few examples and induce reconstruction features suitable for new examples. In the presence of a small number of clean-noise image pairs, for image denoising methods, there will be a lack of a large number

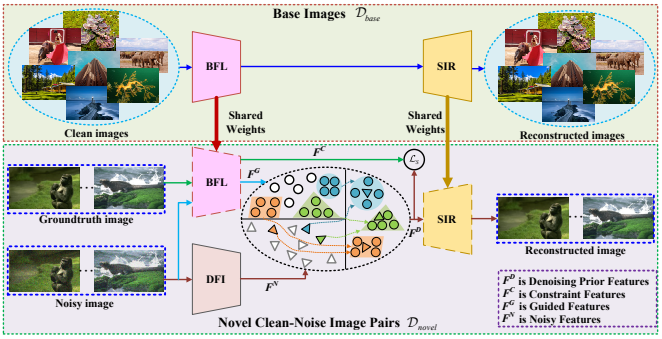

Fig. 1. Overview of our framework for few-shot image denoising. We aim to obtain a few-shot denoising model by using a BFL, a DFI and a SIR. The guided features (*i.e.,* circles) extracted by BFL and the noisy features (*i.e.,* triangles) of DFI are induced to the most relevant latent feature space (the space indicated by the dashed arrow) for generating the denoising prior features. Finally, the obtained denoising prior features are fed into the SIR for image reconstruction.

of prior features for denoising reconstruction in the process of denoising, resulting in a significant decrease in image denoising performance. To overcome the problem of limited denoising reconstructing prior features due to few samples, inspired by few-shot image classification methods [11], we propose a two-stage Few-Shot Learning for Image Denoising (FSLID). As shown in Fig. 1, given the base dataset $\mathcal{D}_{base}$ consists of clean images of sufficient quality and easy access, while the novel dataset $\mathcal{D}_{novel}$ consists of only a few clean-noise image pairs. Specifically, **In the first stage**, a reconstructed feature map of a clean image is built using a given base image dataset $\mathcal{D}_{base}$. In detail, the Basic Feature Learner (BFL) and Shared Image Reconstructor (SIR) jointly learn clean image features from $\mathcal{D}_{base}$. Detailedly, an unsupervised training strategy is applied to jointly train the BFL and SIR using only the clean images $\mathcal{D}_{base}$ rather than the clean-noise image pairs, *i.e.*, a clean image in $\mathcal{D}_{base}$ is both input and its own ground truth. *Moreover, the BFL and SIR in this stage only need to be trained once, and they can be reused in the subsequent training stage without retraining.* **In the second stage**, given a novel dataset $\mathcal{D}_{novel}$, the reconstructed feature map of the above clean image is used to induce and constrain to produce new prior features for adaptively removing noise from noisy images. Further, we use the BFL with frozen weights to extract the guided features and constraint features from $\mathcal{D}_{novel}$. At the same time, the proposed Denoising Feature Inducer (DFI) encodes the noisy features from noisy images in $\mathcal{D}_{novel}$. Finally, inducing both the guided features and noisy features, DFI can generate the denoising prior features for the SIR with frozen weights to adaptively denoise the noisy images.

(2) How to redesign an efficient feature extraction module with a simple structure, low computational complexity, and a small number of parameters under the condition of a limited number of feature channels. We propose Multi-Scale Feature Recursive (MSFR) to modularly formulate an efficient Denoising Feature Inducer (DFI) to capture more diverse contextual feature information under a limited number of feature channels. Furthermore, our MSFR is a refined, low-channel number, recursive multi-branch structure. Due to these characteristics, the proposed MSFR can significantly reduce the number of model parameters and computational complexity.

Our proposed FSLID outperforms competitive baseline methods on multiple datasets and in various settings. Besides, it also demonstrates good transferability from the base image dataset to another different novel noisy image dataset. In summary, the contributions of our work are:

- The proposed two-stage Few-Shot Learning for Image Denoising (FSLID) method aims to achieve effective denoising performance using a limited number of noisy-clean image pairs. In this method, the first stage consists of a Basic Feature Learner (BFL) and a Shared Image Reconstructor (SIR), while the second stage is built with a Denoising Feature Inducer (DFI).
- In the first stage, the BFL and SIR work in conjunction to learn clean image features from the initial population dataset $\mathcal{D}_{base}$. This joint learning approach allows the FSLID method to make full use of the information contained in the noisy-clean image pairs, effectively reducing the demand for a large number of noisy-clean image pairs.
- The second stage employs a Multi-Scale Feature Recursive (MSFR) approach to form a Denoising Feature Inducer (DFI) capable of capturing diverse contextual feature information with a limited number of feature channels. The MSFR approach modularly formulates the DFI, reducing the number of model parameters and computational complexity while preserving the ability to capture rich contextual information.

II. Related Work

Image Denoising In the last few years, significant progress has been made in deep neural networks on image denoising tasks[1], [12], [13], [14]. DnCNN [15] introduces batch normalization and residual learning in deep neural networks for image denoising, and its performance is much better than traditional image denoisers. One early example is FFDNet [16], which utilizes artificially set noise levels to effectively remove non-uniform noise in images. Building upon this, RDN [17] combines densely connected blocks and global residual learning to effectively fuse local and global features for image denoising, achieving even better performance. Another approach is RIDNet [18], which enhances the flow of high-frequency information through an attention mechanism and utilizes a residual structure to reduce the transfer of low-frequency information, thereby reconstructing detailed information in denoised images. Furthermore, CBDNet [19] takes a different approach by employing a noise estimation sub-network that predicts noise maps as prior knowledge to remove real noise in real-world scene images, achieving state-of-the-art performance. Subsequently, numerous designed modules were further used to improve image denoising performance, such as deformable convolution (*e.g.*, SANet [20]) and attention block (*e.g.*, EFF-Net [21]). Although these excellent denoising methods have state-of-the-art performances, they rely heavily on datasets composed of a large number of clean-noise image pairs. Collecting a large amount of the training dataset is challenging and costly, severely limiting the effectiveness of denoisers.

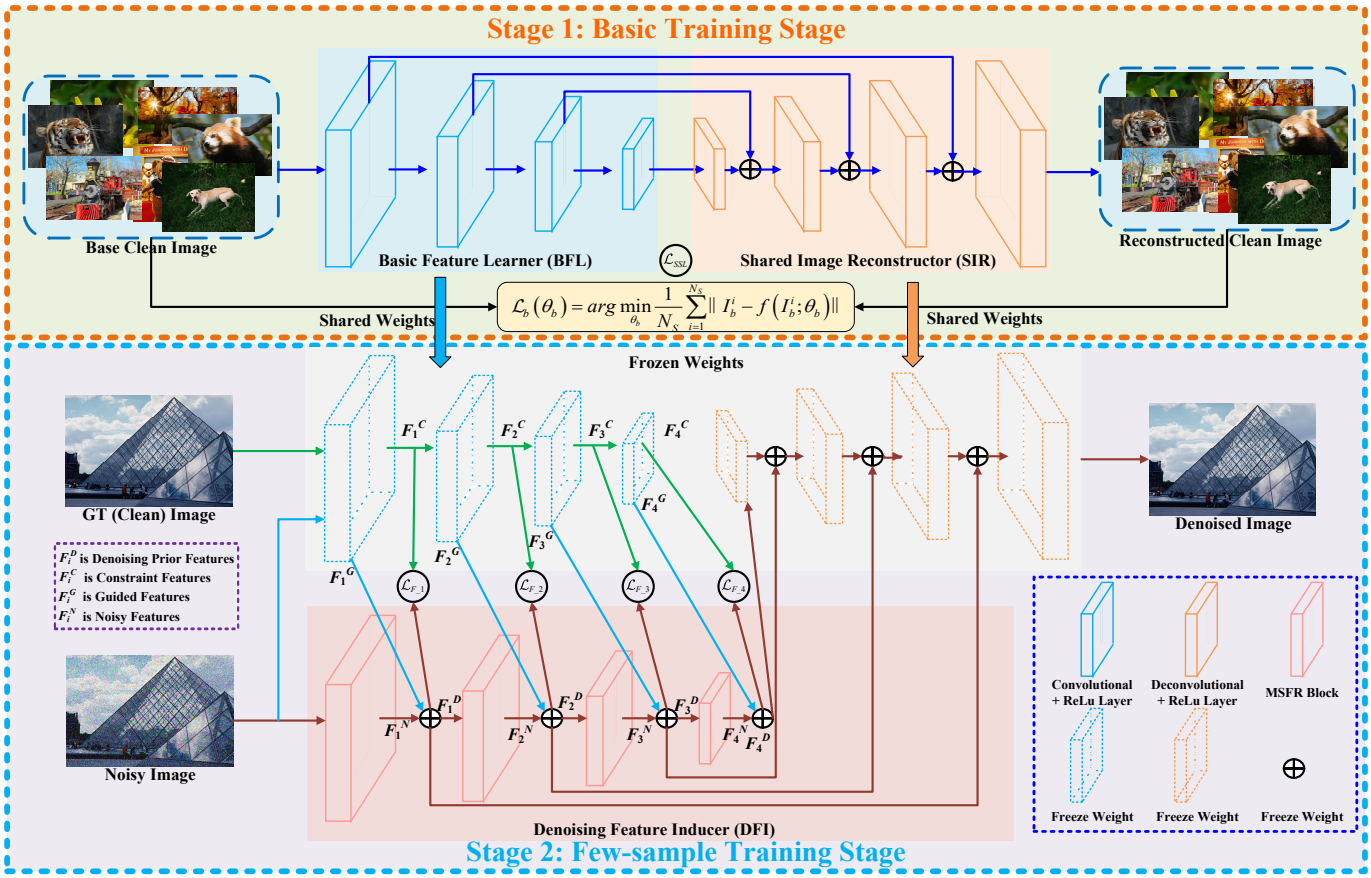

Fig. 2. Schematic of FSLID. It consists of a BFL, a DFI, and a SIR. In the first stage, BFL and SIR jointly learn to express the diverse features of clean images in $\mathcal{D}_{base}$. In the second stage, DFI utilizes the Multi-Scale Feature Recursive (MSFR) module to first encode the noisy features for noisy images in $\mathcal{D}_{novel}$. Then, inducing both the guided features and noisy features, DFI can generate the denoising prior features for the SIR with frozen weights to denoise the noisy images adaptively.

Few-shot Learning The few-shot learning problem aims to learn transferable knowledge across different tasks with only a few novel images [22], [23], [24], [25]. The difficulty lies in the generalization to the unknown feature space while remaining the accuracy as much as possible, which is typically the weigh. Li et al. [26] adopts a one-shot learning training strategy, which classifies novel images through knowledge obtained from the pre-trained model through Bayesian inference. Koch et al. [27] show that the Siamese architecture can beat multiple classification baselines in the k-shot image binary verification task. Luo et al. [28] use the transfer learning to adaptively learn the feature from novel class of images and classify them. Currently, although there is a lack of research on using few-shot learning for image denoising, some attempts to address this problem have emerged. For example, FSMD [29] uses a meta-training approach to fine-tune a known synthetic noise model using a small sample dataset to achieve the goal of few-shot image denoising. However, This is limited by the known synthetic noise model, *i.e.*, the performance of the known synthetic noise model determines the performance after fine-tuning with a small sample dataset. MNC-Net [30] addresses the problem of limited reconstruction features due to a lack of samples by acquiring noise features from a large number of noisy images (without corresponding clean images) through contrastive learning and incorporating them into the image reconstruction network for few-shot image denoising. Specifically, the propsed **FSLID** in this paper and Multilevel Noise Contrastive Network (**MNC-Net**) have different *application scenarios* and *technical approaches* to address the challenge of limited reconstruction features in image denoising. ***Different application scenarios:*** MNC-Net is suitable for scenarios where there is a large number of noisy images without corresponding clean images, while proposed FSLID in this paper is designed for scenarios with a large number of clean images. ***Different technical approaches:*** (a) In MNC-Net, the image reconstruction network incorporates noise feature maps acquired from a large number of noisy images through contrastive learning to reconstruct denoised images, while FSLID builds a reconstructed feature map of a clean image in the first stage and then uses the reconstructed feature map to produce new prior features for removing noise in the second stage. (b) FSLID uses Multi-Scale Feature Recursion (MSFR) to form an efficient Denoising Feature Inducer (DFI) that can capture more diverse contextual feature information with a limited number of feature channels. MSFR is a refined, low-channel-count, recursive multi-branch structure, which

significantly reduces the number of model parameters and computational complexity. MNC-Net does not have a similar technique.

The proposed FSLID overcomes the challenges faced by traditional deep neural network-based image denoising and few-shot learning methods in scenarios where only a limited number of samples are available to reconstruct the clean image features. Our FSLID starts by establishing a reconstructed feature map of a clean image using the given base image dataset $\mathcal{D}_{base}$. In the second stage, this reconstructed feature map is used to induce new prior features and adaptively remove noise from novel images in $\mathcal{D}_{novel}$. To achieve an efficient Denoising Feature Inducer (DFI), the work introduces Multi-Scale Feature Recursion (MSFR), which enables the capture of diverse contextual features with a limited number of feature channels, thereby reducing the model parameters and computational complexity.

III. Problem Setup

In this work, a novel and realistic setting is defined for few-shot image denoising, in which two types of datasets can be used for training, *i.e.*, the base image dataset $\mathcal{D}_{base}$ and the novel image dataset $\mathcal{D}_{novel}$. For the $\mathcal{D}_{base}$, it contains a huge number of available high-quality clean images. $\mathcal{D}_{novel}$ consists of only a few clean-noise image pairs. The above dataset settings are set according to the practical situation. This is because in real scenes, a very limited number of clean-noise image pairs can be collected. In other words, it is very difficult to obtain clean-noise image pairs on a large scale. Hence, solving this few-shot image denoising problem is heavily pressing.

As already explained, the few-shot image denoising network has two stages and two corresponding datasets. Here, in the first stage, we define $\mathcal{D}_{base} = \{(I_b)\}$ as the training dataset of sufficient high-quality *base clean images*, where $I_b \in \mathbb{R}^{H\times W}$ is a clean image. In the second stage, we define $\mathcal{D}_{novel} = \left\{\left(I_n^i, G_n^i\right)\right\}_{i=1}^{K}$ as the training set of *novel clean-noise image pairs* employed during the second stage, where $G_n^i \in \mathbb{R}^{H\times W}$ is the corresponding ground truth for noisy image I_n^i, and K denotes the number of clean-noise image pairs randomly drawn from $\mathcal{D}_{novel}$ (K = 20, 40 and 60 in benchmarks). Importantly, the training datasets $\mathcal{D}_{base}$ and $\mathcal{D}_{novel}$ are disjoint, *i.e.*, $\mathcal{D}_{base} \cap \mathcal{D}_{novel} = \varnothing$.

IV. Method

In this section, we first describe the Basic Feature Learner (BFL) in §IV-A, then introduce the Shared Image Reconstructor (SIR) in §IV-B and the Denoising Feature Inducer (DFI) in §IV-C. Finally, we elaborate on the learning scheme of Few-Shot Image Denoising Network (FSLID) in §IV-D.

A. Basic Feature Learner (BFL)

To fully learn the representations of clean images in $\mathcal{D}_{base}$, we propose a Basic Feature Learner (BFL). The essence of the BFL is an encoder structure composed of convolutional layers and PReLU activation layers, as shown in Fig. 2. Formally, the BFL and the SIR in the proposed framework form a typical *U*-shaped network structure. We adopt the characteristics of this structure to encode high-quality clean base images and embed them into the feature latent space. Then, the SIR reconstructs the base images from the feature latent space to train the BFL to learn the representations of the base images. Therefore, the equation of the features extracted by the BFL is expressed as Eqn. 1:

$$F_1, F_2, F_3, F_4 = f_B\left(I_b; \theta_B\right), \tag{1}$$

where F_1, F_2, F_3 and F_4 represent the features of different scales of clean base images, f_B denotes the function of the BFL, and θ_B is the parameters of the BFL. I_b ($I_b \in \mathcal{D}_{base}$) indicates clean images.

B. Shared Image Reconstructor (SIR)

To reconstruct high-quality images from the latent feature space, we propose Shared Image Reconstructor (SIR) method. Such a shared structure may not only reduce the parameters of the entire network, but also reduce the cost of network reasoning. The essence of SIR is a decoder structure composed of deconvolution layers and PReLU activation layers, as shown in Fig. 2. Formally, both BFL-SIR and DFI-SIR can form a U-shaped network. Therefore, the equation representing the features extracted by SIR is expressed as Eqn. 2:

$$I = f_S\left(F_1^{in}, F_2^{in}, F_3^{in}, F_4^{in}; \theta_S\right), \tag{2}$$

where F_1^{in}, F_2^{in}, F_3^{in} and F_4^{in} represent the features with different scales extracted from BFL or DFI, f_S denotes the function of the SIR, and θ_S is the parameters of the SIR.

C. Denoising Feature Inducer (DFI)

To induce the multi-scale guided features and noisy features into generating denoising prior features, we propose Denoising Feature Inducer (DFI). DFI is an encoder mainly composed of the Multi-Scale Feature Recursive (MSFR) modules (*i.e.*, the core component of DFI is MSFR), as shown in Fig. 3.

The MSFR block is constructed by the Conventional Separators (CS). The CS first uses a 3×3 convolution layer, including the activation function and split function, to extract input feature maps for multiple subsequent CS steps. For each step, the split function is adopted to divide the channels of extracted feature maps into two equal parts. One part is retained and the other part is fed into the next CS step. Then, all the feature maps are concatenated together. The above process is shown as Eqn. 3:

$$\begin{aligned} &F_i^{rest}, F_i^{next} = Split_i\left(F_i\right), \\ &F^s = Concat_{i=1}^{m}\left(F_i^{rest}, Split_i\left(F_i^{next}\right)\right), \end{aligned} \tag{3}$$

where F_i ($F_i \in \mathbb{R}^{C\times H\times W}, C = n$) is the i^{th} ($i \in \{1, 2, ..., m\}$) input feature map, m represents the number of splits. $Split_i$ is the CS operation, and $Concat$ denotes the concatenation operation at the channel extent. F_i^{rest} ($F_i^{rest} \in \mathbb{R}^{C\times H\times W}, C = \frac{n}{2^{i-1}}$) represents the retained feature, while F_i^{next} ($F_i^{next} \in \mathbb{R}^{C\times H\times W}, C = \frac{n}{2^{i-1}}$) indicates the feature for the next CS operation. F^s ($F^s \in \mathbb{R}^{C\times H\times W}, C = n$) is the feature output at this step.

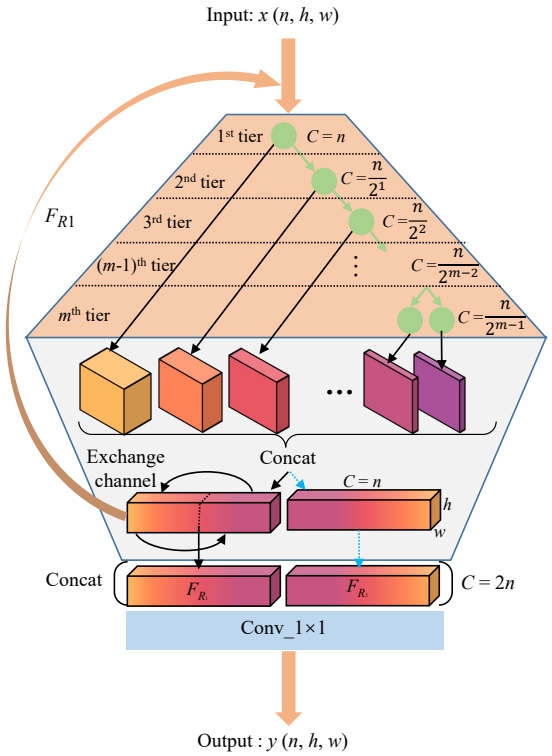

Fig. 3. Multi-Scale Feature Recursive (MSFR) block structure. The light-orange background area (top) represents the internal structure of the block, and the green circle represents the convolutional separator (*i.e.*, *conv* + *prelu* + *split* operation). The blue arrow indicates the feature F_{R_2} generated after recursion. The gray background area denotes the feature integration process.

To further obtain the feature maps with more abundant extraction levels, we divide the integrated feature maps into two equal parts and reverse feature maps at the channel axis, producing the feature maps F_{R_1} ($F_{R_1} \in \mathbb{R}^{C\times H\times W}, C = n$). Then, F_{R_1} is recursively fed into the CS operations again to generate the feature maps F_{R_2} ($F_{R_2} \in \mathbb{R}^{C\times H\times W}, C = n$) with abundant extraction levels. Finally, a 1×1 convolutional layer is employed to fuse and reduce the channels of concatenation between F_{R_1} and F_{R_2}. The above process is shown as Eqn. 4:

$$F_{out} = f_{1\times 1}\left(Concat\left(F_{R_1},\ F_{R_2}\right)\right), \tag{4}$$

where F_{out} ($F_{out} \in \mathbb{R}^{C\times H\times W}, C = n$) is the feature output at this stage. $f_{1\times 1}$ denotes the convolutional layer with kernel size of 1×1.

The MSFR can effectively generate the features with abundant levels under the condition that the number of channels is unchanged, so that the prior features required for noise removal can be induced more easily.

D. Learning Scheme Details

In order to promote the generalization performance of the FSLID, unlike the conventional image denoising model training scheme, we have developed a novel two-stage learning scheme.

First stage: unsupervise training on $\mathcal{D}_{base}$. We adopt the auto-encoder-like setting, *i.e.*, the input image is the supervised signal of the predicted (output) image, to jointly train the BFL and SIR on $\mathcal{D}_{base}$. This not only avoids collecting or simulating ground truth for the input images, but also conforms to the conditions in the real application scene, because the number of images with ground truth in the real scene is extremely few. Therefore, we optimize the U-shaped network composed of BFL and SIR by minimizing the following loss:

$$\mathcal{L}_b\left(\theta_b\right) = arg\ \min_{\theta_b} \frac{1}{N_S}\sum_{i=1}^{N_S} \|I_b^i - f\left(I_b^i; \theta_b\right)\|, \tag{5}$$

where $f : \mathbb{R}^{H\times W} \rightarrow \mathbb{R}^{H\times W}$ is the U-shaped network function with the vector of parameters θ_b to be optimized. The H and W are the height and width of an image. N_S denotes the number of base clean images.

Second stage: few-sample training on $\mathcal{D}_{novel}$. Since the BFL and the SIR are jointly trained together at the first stage, the encoding-decoding distribution is consistent. Because the decoder SIR is shared in both two training stages, in order to be consistent with the decoding distribution of the decoder at the second stage, the DFI and the SIR with frozen weights form another U-shaped network [31] are jointly trained on $\mathcal{D}_{novel}$.

Concretely, given a training set $\mathcal{D}_{novel}$ with clean-noise image pairs, the clean image is fed into the BFL with frozen weights to extract the constraint features F_i^C (*i.e.*,$F_i^C \in \mathcal{F}^C, i = \{1, ..., M\}$) in the latent space. At the same time, the noise image is used as the input data of the DFI and the BFL with frozen weights to generate and embed corresponding noisy features F_i^N (*i.e.*,$F_i^N \in \mathcal{F}^N, i = \{1, ..., M\}$) and guided features F_i^G (*i.e.*,$F_i^G \in \mathcal{F}^G, i = \{1, ..., M\}$) into the common latent space, respectively. The above process is shown by the green arrow and the dark red arrow in Fig. 2. To enable DFI to induce guided features and noisy features in the common latent feature space into generating denoising prior features, we use constraint features to constrain the features learned from each MSFR block in DFI to generate the denoising prior features. Finally, denoising prior features are fed into the SIR to reconstruct the denoised images. In the above process, each corresponding loss function (*i.e.*, $\mathcal{L}_{F_1}$, $\mathcal{L}_{F_2}$, $\mathcal{L}_{F_3}$ and $\mathcal{L}_{F_4}$) is calculated according to Eqn. 6.

$$\mathcal{L} = \frac{1}{M}\sum_{i}^{M}\sum_{j=1}^{C_i} \|F_{i,j}^c - F_{i,j}^n\|, \tag{6}$$

where M ($M = 4$) is the number of extractors (or layers) to obtain the feature maps $\mathcal{F}^n$ and $\mathcal{F}^c$. C_i represents the number of channels of feature maps. The overall loss function of the second stage is shown in Eqn. 7.

$$\mathcal{L}_s = \mathcal{L}_{F_1} + \mathcal{L}_{F_2} + \mathcal{L}_{F_3} + \mathcal{L}_{F_4}, \tag{7}$$

where $\mathcal{L}_s$ is the overall loss function of the second stage.

V. Experiments

In this section, we introduce the implementation details of the proposed FSLID, experimental settings, and few-shot image denoising benchmark in §V-A, §V-B and §V-C. Then, in §V-D, we employ different datasets to evaluate the proposed FSLID and various baselines. In addition, we also provide various

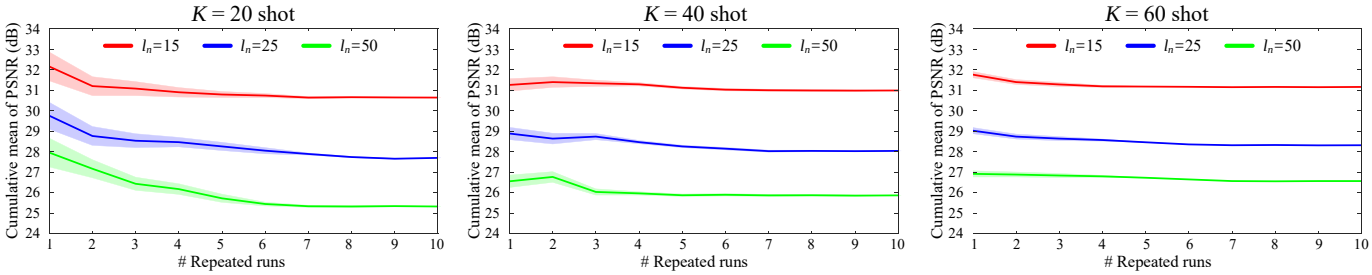

Fig. 4. Cumulative mean of PSNR with 95% confidence intervals across 10 repeated runs, tested on CBSD68 dataset under training using 10 different groups of random selected training samples on various training shots for the second stage, *i.e.*, $\mathcal{D}_{novel}^{S,K=20}$, $\mathcal{D}_{novel}^{S,K=40}$ and $\mathcal{D}_{novel}^{S,K=60}$ dataset. The means and variances become stable after around 6 runs.

TABLE II. Parameter settings in the MSFR block.

m^{th} tier	1	2	3	4
Number of channels	64	32	16	8
Kernel size in CS uint	3×3	7×7	5×5	3×3

ablation studies and visualizations in §V-E. Finally, we discuss the limitations of the FSLID in §V-F.

A. Implementation Details

Inference Firstly, the guided features and noisy features are respectively extracted by BFL and DFI from a validation noisy image. Then, they are induced to generate the denoising prior features. Finally, the obtained denoising prior features are fed into the SIR for obtaining the corresponding denoised image.

Hyper-parameter Settings The proposed FSLID is composed of a BFL, a DFI, and a SIR. In the first training stage, the BFL is composed of 4 convolutional layers with a size of 3×3 and 64 convolution kernels. The SIR consists of 4 deconvolution layers with 64 deconvolution kernels. In the second training stage, the DFI is mainly composed of 4 MSFR blocks. There are 4 tiers in each MSFR block (*i.e.*, $m = 4$), which means there are 4 CS units. Meanwhile, the input channel of the BFL and DFI is set to $n = 64$. The specific parameters of the MSFR block are shown in Table II.

Our FSLID is implemented by Pytorch [1] and trained on four Nvidia RTX TITAN GPUs with a batch size of 20. We set a total of 1000 epochs, including two training stages, *i.e.,* the 1^{st} to 200^{th} epochs are the first training stage, and the remaining 800 epochs are the second training stage. In the first training stage, we extract patches with a size of 128×128 from the base clean images in the $\mathcal{D}_{base}$ dataset, which are used as the input of the BFL. In the second stage, we randomly extract the corresponding image patches with a size of 128×128 from the ground truth images and the noise images in the $\mathcal{D}_{novel}$ dataset. The noise image patches and the corresponding ground truth image patches are respectively used as the input of the DFI and the BFL with frozen weights. Adam [32] is used as optimizer, in which β_1 and β_2 adopt default values of 0.9 and 0.999, respectively. The initial learning rate is 1×10^{-3}. The FSLID's parameters are initialized using the *Kaiming* method in [33].

[1]https://pytorch.org/

B. Experimental Setup

Noisy Image Datasets In the first training stage, we select the widely used DIV2K [34] dataset as the training data (base image set $\mathcal{D}_{base}$) of the BFL and SIR, including only $2,650$ high-resolution clean images. The BFL and SIR in the first stage only need to be trained once, and are reused in the subsequent training stage without retraining. For synthetic noise image and real noise image experiments, we use CBSD400 (400 image pairs) [35] and SIDD (160 image pairs) dataset [36] as novel synthetic clean-noise image pairs set $\mathcal{D}_{novel}^{S}$ and novel real clean-noise image pair set $\mathcal{D}_{novel}^{R}$, respectively. In the second training stage, for the synthetic noise image experiment, we randomly select $K = 20, 40, 60$ clean-noise image pairs from $\mathcal{D}_{novel}^{S}$ as training data. In addition, different numbers K of novel clean-noise image pair sets $\mathcal{D}_{novel}^{S,K}$ contain three levels of noise, such as additive Gaussian noise with a level of $l_n = 15, 25, 50$. Similarly, for the real noise image experiment, we also randomly select $K = 20, 40, 60$ clean-noise image pairs from $\mathcal{D}_{novel}^{R}$ as training data. Since the SIDD data set contains images with a large resolution, we choose to obtain the patch by the center cropping method, *i.e.*, only one patch is obtained for an image.

Evaluation Metrics For synthetic images and real-world images, we calculate PSNR, SSIM [40], and LPIPS [41] for the different methods. PSNR and SSIM focus on the fidelity of the image rather than visual quality, while LPIPS pays more attention to the similarity of the visual features. The value of PSNR and SSIM is positively correlated with visual quality, while the value of LPIPS is negatively correlated with visual quality. That is, the larger the PSNR and SSIM values are, the better the visual effect is, and the smaller the LPIPS value is, the closer the predicted denoising images to the ground truth images are.

C. Few-shot Image Denoising Benchmark

Benchmark Due to the small number of samples used for the second training stage, the randomly selected samples have large variance [42]. This makes it difficult to draw valid conclusions from comparisons with other methods, because the

TABLE III. Proposed FSLID method comparisons with four baseline models (*i.e.*, HINet-ft, MIRNet-ft, HINet-ft-full, and MIRNet-ft-ful). Average PSNR of the denoised synthetic grayscale images from BSD68, KODAK and Set12 datasets. The values of PSNR are positively correlated with visual quality. The computational efficiency (FLOPs) is obtained on the patch size of 256×256.

Methods	Noise Level	Parameters	FLOPs	K=20			K=40			K=60		
				BSD68	KODAK	Set12	BSD68	KODAK	Set12	BSD68	KODAK	Set12
HINet-ft	l_n = 15	88.67M	42.68G	28.60	28.92	26.05	28.73	29.03	26.19	29.15	29.21	26.63
MIRNet-ft		4.29M	138.88G	26.98	27.45	25.75	27.36	27.71	26.28	27.54	27.82	26.35
HINet-ft-full		88.67M	42.68G	29.24	29.65	27.63	29.56	30.16	27.84	29.93	30.37	27.93
MIRNet-ft-full		4.29M	138.88G	28.88	29.63	26.86	29.36	29.88	27.24	29.67	30.00	27.51
FSLID (Ours)		1.34M	5.92G	**30.63**	**30.76**	**29.03**	**30.98**	**31.23**	**29.23**	**31.14**	**31.40**	**29.46**
HINet-ft	l_n = 25	88.67M	42.68G	23.26	24.33	22.52	23.33	24.59	22.89	23.54	24.80	22.99
MPRNet-ft		4.29M	138.88G	23.11	23.97	22.44	23.47	24.35	22.92	23.57	24.56	23.01
HINet-ft-full		88.67M	42.68G	27.22	28.08	25.53	27.31	28.23	25.82	27.41	28.46	26.01
MIRNet-ft-full		4.29M	138.88G	25.64	26.51	23.97	26.07	26.87	24.31	26.16	27.06	24.39
FSLID (Ours)		1.34M	5.92G	**27.69**	**28.22**	**26.54**	**28.02**	**28.63**	**26.81**	**28.31**	**28.90**	**27.14**
HINet-ft	l_n = 50	88.67M	42.68G	17.80	18.99	17.31	18.24	19.06	17.85	18.34	19.25	18.11
MIRNet-ft		4.29M	138.88G	17.47	18.42	17.11	17.92	18.81	17.57	18.03	19.02	17.66
HINet-ft-full		88.67M	42.68G	23.25	23.63	22.54	23.82	24.06	23.06	24.02	24.15	23.24
MIRNet-ft-full		4.29M	138.88G	21.84	21.99	20.66	22.32	22.42	20.93	22.46	22.53	21.14
FSLID (Ours)		1.34M	5.92G	**25.32**	**26.01**	**24.12**	**25.86**	**26.22**	**24.39**	**26.56**	**27.08**	**24.93**

TABLE IV. Proposed FSLID method comparisons with four self-supervised models (*i.e.*, N2N [37], N2V [38], N2S [39], and NLM [39]). Average PSNR of the denoised synthetic grayscale images from BSD68, KODAK and Set12 datasets. The values of PSNR are positively correlated with visual quality. The computational efficiency (FLOPs) is obtained on the patch size of 256×256. † denotes that the network structure of N2N and N2V is a custom U-Net [31] architecture [38]. ‡ indicates that the network structure of N2S and NLM is a U-Net [31] architecture with two layers [39].

Methods	Parameters	FLOPs	l_n = 15			l_n = 25			l_n = 50		
			BSD68	KODAK	Set12	BSD68	KODAK	Set12	BSD68	KODAK	Set12
N2N†	1.41M	43.66G	30.22	30.67	29.07	28.91	30.26	28.29	25.11	25.64	24.26
N2V†	1.41M	43.66G	29.97	30.61	28.91	28.58	29.71	27.62	25.05	25.67	24.21
N2S‡	4.16M	64.43G	30.13	30.48	29.20	28.82	29.84	27.98	25.25	25.78	24.29
NLM‡	4.16M	64.43G	28.90	29.25	27.87	27.51	28.14	26.42	24.21	24.83	23.45
FSLID (Ours) K=20	1.34M	5.92G	30.69	30.84	28.93	29.21	29.73	28.11	25.24	25.88	24.18
FSLID (Ours) K=40	1.34M	5.92G	30.98	31.43	29.43	29.50	30.22	28.40	25.84	26.30	24.47
FSLID (Ours) K=60	1.34M	5.92G	**31.35**	**31.61**	**29.67**	**29.91**	**30.50**	**28.74**	**26.65**	**27.17**	**25.02**

performance differences may not be credible. To address the above issues, we built a few-shot image denoising benchmark. The model is trained for multiple runs using different groups of random selected training samples to produce the means and confidence intervals for PSNR. In Fig. 4, we show the average cumulative PSNR of 10 repeated runs with $K = 20, 40, 60$ for the second-stage training process. From this figure, with the increasing replicate runs, the mean PSNRs gradually stabilize and the confidence intervals get smaller on all cases of using different training shots, allowing for more effective comparisons. Therefore, in the following experiments, we train the proposed model for 10 runs using different groups of random selected samples for all training shots to generate a stable metric mean.

Baseline We compare our FSLID with four competitive baselines. These four baselines are mainly based on HINet[43] and MIRNet [44] denoisers, all of which can be learned through a two-stage few-shot training strategy. The four baseline models are divided into two categories. The first type of baseline model is only trained by using the same number of novel clean-noise image pair sets $\mathcal{D}^K_{novel}$ as in our second training stage. For this model, we term this baseline as HINet-ft and MIRNet-ft. Another type of baseline model uses two training stages, *i.e.*, the baseline model first uses the base image set $\mathcal{D}_{base}$ for pre-training in the first training stage. Secondly, in the second training stage, this baseline model employs all samples in novel clean-noise image pair set $\mathcal{D}^K_{novel}$ for image denoising learning. We term such baselines as HINet-ft-full and MIRNet-ft-full. Comparing with these baselines can help understand the advantages of few-shot learning brought by our proposed FSLID.

D. Comparisons

In this section, we evaluate and compare the proposed FSLID with baseline models. In synthetic noisy experiments, we present our main results on novel clean-noise image pair dataset (*i.e.*, commonly used evaluation CBSD68, KODAK and Set12 datasets) with adding different levels of noise ($l_n = 15, 25, 50$). Subsequently, in the real-world noisy experiments, SIDD dataset is adopted to evaluate the compared methods.

Synthetic Grayscale Noisy Images. In the comparisons of grayscale noisy images, we compare the proposed FSLID with four baseline models (*i.e.*, HINet-ft, MIRNet-ft, HINet-ft-full, and MIRNet-ft-full). Table III shows the average results of PSNR on BSD68, KODAK24, and Set12 grayscale image datasets with three different noise levels under training with the $\mathcal{D}^{S,K=20}_{novel}$, $\mathcal{D}^{S,K=40}_{novel}$ and $\mathcal{D}^{S,K=60}_{novel}$ dataset. From this table, our FSLID achieves the best results compared to the four baseline

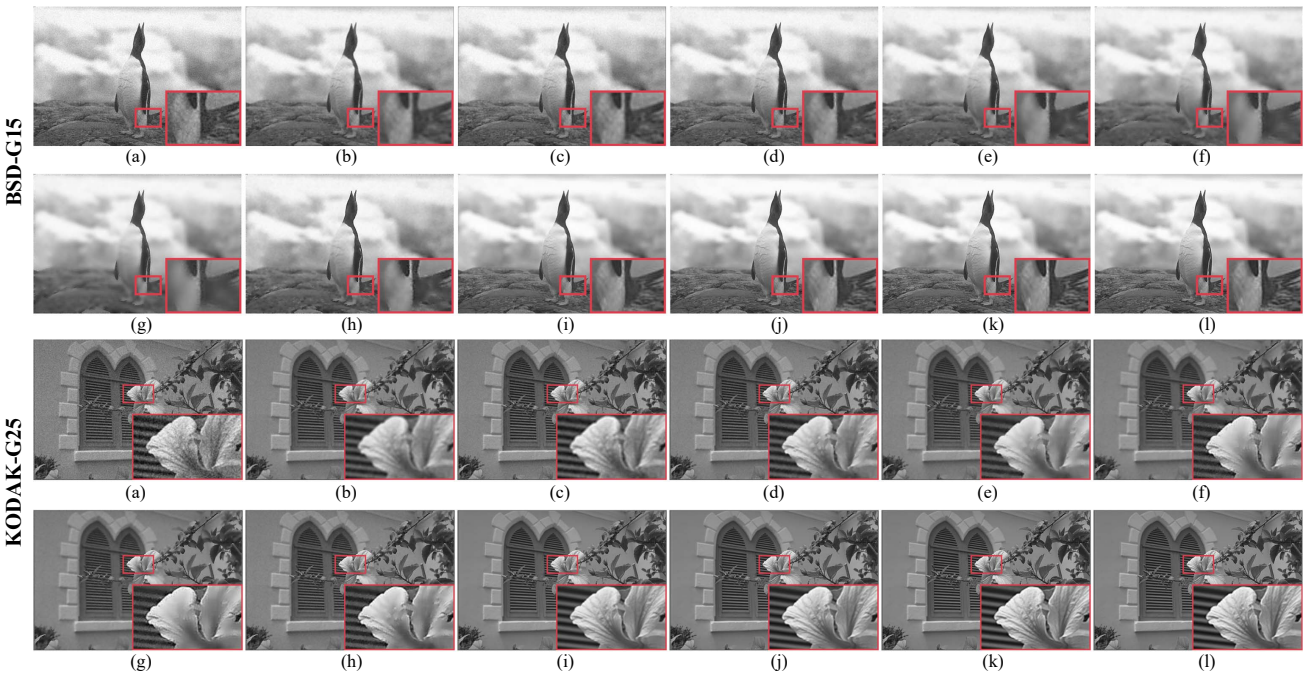

Fig. 5. Comparisons of visual results on different grayscale image datasets. The first two rows represent the visual comparison of different methods on the BSD68 dataset with $l_n = 15$. The last two rows represent the visual comparison of different methods on the KODAK dataset with $l_n = 25$. Please zoom in for a better view. (a) Noisy input. (b) MIRNet-ft. (c) HINet-ft. (d) MIRNet-ft-full. (e) MIRNet-ft-full. (f) N2N. (g) N2V. (h) N2S. (i) NLM. (j) FSLID (Ours) K=20. (k) FSLID (Ours) K=40. (l) FSLID (Ours) K=60.

models. Additionally, in terms of computational complexity FLOPs, the FSLID is reduced by approximately **7.2 times** and **23.5 times** compared to HINet-ft and MIRNet-ft, respectively. Furthermore, in terms of the number of parameters, the FSLID is reduced by approximately **66.2 times** and **3.2 times** compared to HINet-ft and MIRNet-ft, respectively. Hence, compared to the baseline models, the FSLID composed of MSFR can significantly reduce the number of model parameters and computational complexity. The performances produced by our FSLID are improved satisfactorily compared to four baseline models (*i.e.*, HINet-ft, MIRNet-ft, HINet-ft-full, and MIRNet-ft-full) and are the best among all the compared methods. To verify the superiority of our framework, we compare the proposed FSLID with self-supervised methods (*i.e.*, N2N [37], N2V [38], N2S [39], and NLM [39]) using few or no paired data pairs, as shown in Table IV. From the Table IV, the propsed FSLID achieves the best results compared to the four self-supervised methods, while the FLOPs of the FSLID are reduced about **7.4 times** than N2N and N2V. The performances produced by our FSLID are improved satisfactorily compared to N2N [37], N2V [38], N2S [39] and NLM [39] and are the best among all the compared methods.

Additionally, more visual comparisons on grayscale image dataset are shown in Fig. 5. Compared with baselines trained on the $\mathcal{D}_{novel}^{S,K=20}$ dataset and four self-supervised methods on different test datasets, our method can recover images more satisfactorily without bringing in apparent artifacts and over-smoothness. This indicates the effectiveness of the proposed FSLID. Furthermore, since these test datasets are independent from the training datasets, these results fully show that the framework of FSLID achieves a strong generalization for denoising the synthetic noisy grayscale images.

Synthetic Color Noisy Images Main results of our FSLID and four baseline models (*i.e.*, HINet-ft, MIRNet-ft, HINet-ft-full and MIRNet-ft-full) are presented on CBSD68, KODAK and Set12 datasets under training with the $\mathcal{D}_{novel}^{S,K=20}$, $\mathcal{D}_{novel}^{S,K=40}$ and $\mathcal{D}_{novel}^{S,K=60}$ dataset in Table V. *It can be noticed from the table that the proposed FSLID significantly outperforms the baselines, especially when the paired images are very scarce.* Moreover, the performance of the proposed FSLID is significantly better than HINet-ft and MIRNet-ft. However, the proposed FSLID has about **7.1 times** less FLOPS than HINet-ft and about **22.9 times** less FLOPS than MIRNet-ft. This indicates the necessity of using two training stages in our framework. At the same time, the performances of HINet-ft-full and MIRNet-ft-full are lower than the FSLID, which shows the superiority of our proposed FSLID. Moreover, the performances of HINet-ft-full, and MIRNet-ft-full are lower than the FSLID, which shows the superiority of our proposed FSLID. To further illustrate the generalization and robustness of the proposed FSLID on the color image dataset, we also conduct a comparative experiment with FSLID and self-supervised methods (*i.e.*, N2N [37], N2V [38], N2S [39] and NLM [39]), as shown in Table VI. From this table, the proposed FSLID significantly outperforms N2N [37], N2V [38], N2S [39], and NLM [39]. However, the proposed FSLID has about **10.6 times** less FLOPS than N2S, and NLM, and about **7.19 times** less FLOPS than N2N and N2V. This shows that our FSLID has better generalization and robustness than the above four self-supervised methods.

Fig. 6 shows the visual effects of the proposed FSLID, baselines trained on the $\mathcal{D}_{novel}^{S,K=20}$ dataset and four self-supervised methods on different test datasets. Compared with

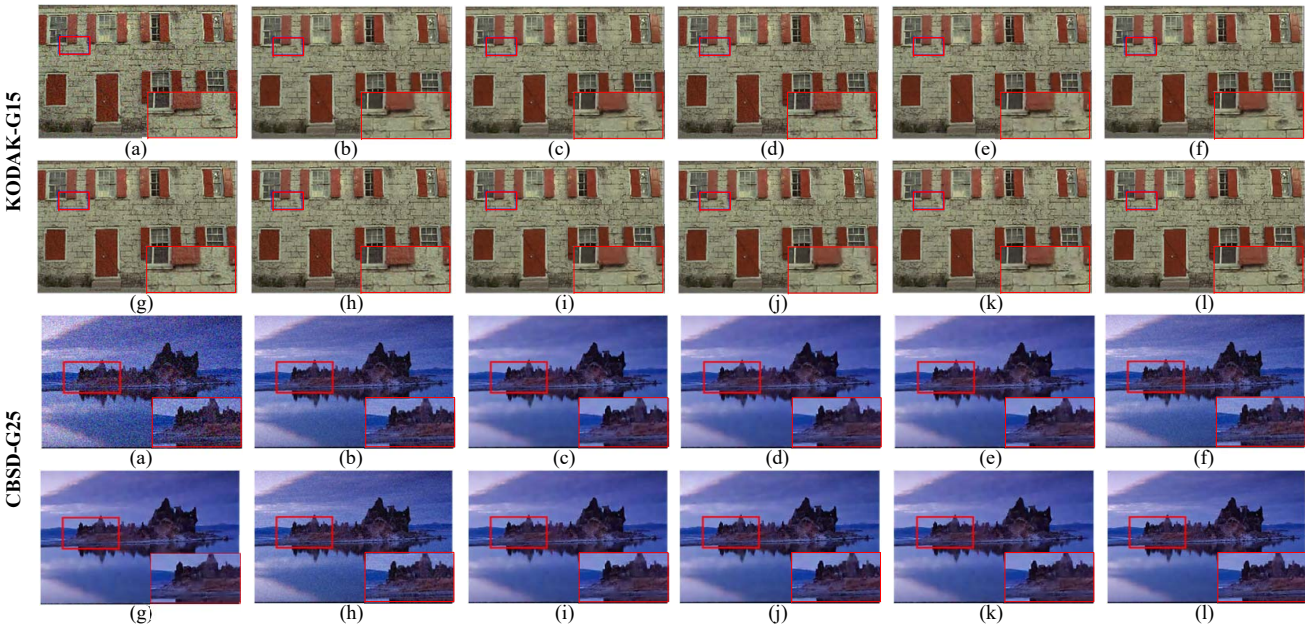

Fig. 6. Comparisons of visual results on the different color image datasets. The first tow rows represent the visual comparison of different methods on the KODAK dataset with $l_n = 15$. The last tow rows represent the visual comparison of different methods on the CBSD68 dataset with $l_n = 25$. Please zoom in for a better view. (a) Noisy input. (b) MIRNet-ft. (c) HINet-ft. (d) MIRNet-ft-full. (e) MIRNet-ft-full. (f) N2N. (g) N2V. (h) N2S. (i) NLM. (j) FSLID (Ours) K=20. (k) FSLID (Ours) K=40. (l) FSLID (Ours) K=60.

TABLE V. Proposed FSLID method comparisons with four baseline models (*i.e.*, HINet-ft, MIRNet-ft, HINet-ft-full, and MIRNet-ft-ful). Average PSNR of the denoised synthetic color images from CBSD68, KODAK and Set12 datasets. The values of PSNR are positively correlated with visual quality. The computational efficiency (FLOPs) is obtained on the patch size of 256×256.

Methods	Noise Level	Parameters	FLOPs	K=20			K=40			K=60		
				CBSD68	KODAK	Set12	CBSD68	KODAK	Set12	BSD68	KODAK	Set12
HINet-ft	l_n = 15	88.72M	42.97G	29.84	30.14	27.32	30.05	30.35	27.51	30.47	30.53	27.95
MIRNet-ft		4.35M	139.16G	28.35	28.71	27.09	28.66	29.01	27.58	28.84	29.12	27.65
HINet-ft-full		88.72M	42.97G	30.54	31.03	28.92	30.94	31.54	29.22	31.31	31.75	29.31
MIRNet-ft-full		4.35M	139.16G	30.32	30.96	28.22	30.73	31.25	28.61	31.04	31.37	28.88
FSLID (Ours)		1.34M	6.07G	**32.02**	**32.17**	**30.26**	**32.21**	**32.66**	**30.66**	**32.57**	**32.83**	**30.89**
HINet-ft	l_n = 25	88.72M	42.97G	24.47	25.63	23.91	24.76	26.02	24.32	24.97	26.23	24.42
MPRNet-ft		4.35M	139.16G	24.33	25.42	23.78	24.73	25.61	24.18	24.83	25.82	24.27
HINet-ft-full		88.72M	42.97G	28.54	29.37	26.96	28.74	29.66	27.25	28.84	29.89	27.44
MIRNet-ft-full		4.35M	139.16G	26.96	27.84	25.38	27.35	28.15	25.59	27.44	28.34	25.67
FSLID (Ours)		1.34M	6.07G	**29.04**	**29.56**	**27.94**	**29.23**	**29.95**	**28.13**	**29.63**	**30.22**	**28.46**
HINet-ft	l_n = 50	88.72M	42.97G	19.13	20.25	18.72	19.62	20.44	19.23	19.72	20.63	19.49
MIRNet-ft		4.35M	139.16G	18.77	19.86	18.33	19.17	20.06	18.82	19.28	20.27	18.91
HINet-ft-full		88.72M	42.97G	24.63	25.07	23.96	25.13	25.37	24.37	25.33	25.46	24.55
MIRNet-ft-full		4.35M	139.16G	23.21	23.41	21.92	23.72	23.82	22.33	23.86	23.93	22.54
FSLID (Ours)		1.34M	6.07G	**26.57**	**27.21**	**25.51**	**27.07**	**27.53**	**25.70**	**27.87**	**28.39**	**26.24**

TABLE VI. Proposed FSLID method comparisons with four self-supervised models (*i.e.*, N2N [37], N2V [38], N2S [39], and NLM [39]). Average PSNR of the denoised synthetic color images from CBSD68, KODAK and Set12 datasets. The values of PSNR are positively correlated with visual quality. The computational efficiency (FLOPs) is obtained on the patch size of 256×256. † denotes that the network structure of N2N and N2V is a custom U-Net [31] architecture [38]. ‡ indicates that the network structure of N2S and NLM is a U-Net [31] architecture with two layers [39].

Methods	Parameters	FLOPs	l_n = 15			l_n = 25			l_n = 50		
			CBSD68	KODAK	Set12	CBSD68	KODAK	Set12	CBSD68	KODAK	Set12
N2N†	1.41M	44.04G	31.64	32.09	30.49	30.33	31.68	29.71	26.53	27.06	25.68
N2V†	1.41M	44.04G	31.28	31.92	30.22	29.89	31.02	28.93	26.36	26.98	25.52
N2S‡	4.19M	64.56G	31.47	31.82	30.54	30.16	31.18	29.32	26.59	27.12	25.63
NLM‡	4.19M	64.56G	30.14	30.49	29.11	28.75	29.38	27.66	25.45	26.07	24.69
FSLID (Ours) K=20	1.34M	6.07G	32.02	32.17	30.26	30.54	31.06	29.44	26.57	27.21	25.51
FSLID (Ours) K=40	1.34M	6.07G	32.21	32.66	30.66	30.73	31.45	29.63	27.07	27.53	25.70
FSLID (Ours) K=60	1.34M	6.07G	**32.57**	**32.83**	**30.89**	**31.13**	**31.72**	**29.96**	**27.87**	**28.39**	**26.24**

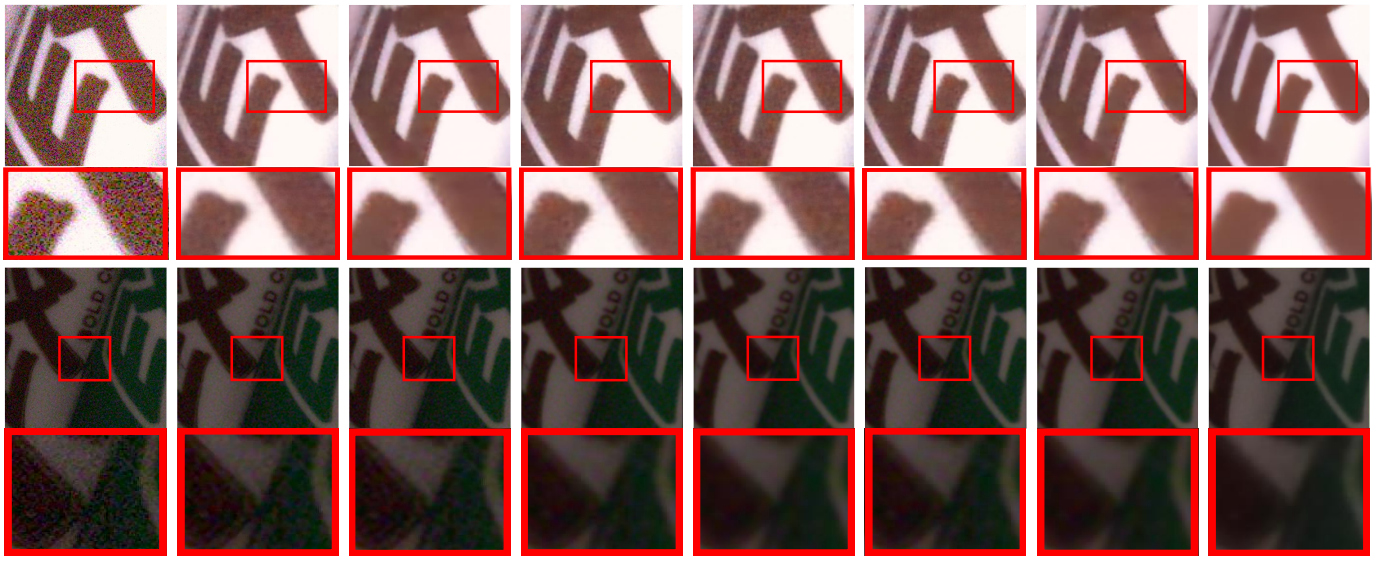

Fig. 7. Visual comparisons between FSLID and its competitors (*i.e.*, four baseline models) in the evaluation of real-noisy image denoising. All test images were from real noisy dataset SIDD. Please zoom in for a better view.

TABLE VII. Comparisons of denoising results using different unsupervised methods on SIDD datasets. The values of PSNR is positively correlated with visual quality. The computational efficiency (FLOPs) is obtained on the patch size of 256×256.

Methods	Parameters	FLOPs	Speed (s)	K=20	K=40	K=60
HINet-ft	88.72M	42.97G	1.35	36.36	37.05	37.27
MIRNet-ft	4.35M	139.16G	0.38	36.18	36.91	37.16
HINet-ft-full	88.72M	42.97G	1.35	37.84	38.16	38.31
MIRNet-ft-full	4.35M	139.16G	0.38	37.32	38.11	38.25
FSID (Ours)	1.34M	6.07G	**0.12**	**38.35**	**38.42**	**38.56**

HINet-ft and MIRNet-ft, after directly using a few clean-noise image pairs for training, we find that HINet-ft and MIRNet-ft cause artifacts and severe tailing in the reconstructed images. This shows that insufficient samples may lead to poor generalization performances of conventional denoisers based on CNNs. In addition, compared with the visual effects of HINet-ft-full and MIRNet-ft-full, the textures and other details in the reconstructed images are excessively smooth. This also shows that even if the training samples increase but do not match the two-stage few-shot training strategy, may causing the under-fitting of the method. Moreover, compared with N2N [37], N2V [38], N2S [39], and NLM [39], we find that the above self-supervised methods produce under-denoising phenomenon (*i.e.*, a small amount of noise remains in the image) in the reconstructed images. The self-supervised denoising method only relies on the noise signal to drive the training process, too much noise as the supervision signal leads to the under-denoising phenomenon in the denoised image. This indicates that the proposed few-shot FSLID method is sufficient and necessary. In summary, the proposed FSLID can obtain competitive quantitative and visual results with a small number of training samples.

Real-World Noisy Images Noise in real images is usually coupled from multiple complex noise sources. Therefore, denoising real-world images is very difficult. This implies, it is persuasive to evaluate denoising methods on real-world noisy datasets. Table VII reports our PSNR/SSIM/LPIPS results on the SIDD test dataset under training on the $K = 20, 40, 60$ of the SIDD train dataset. It is notable that the FSLID produces the best results compared to the HINet-ft, MIRNet-ft, HINet-ft-full and MIRNet-ft-full on the SIDD test dataset evaluations, while using much fewer parameters and FLOPs. Additionally, the inference time of our FSLID using the patch size of 256×256 is the smallest among all the evaluation methods. Compared with HINet-ft-full, and MIRNet-ft-full, the inference speed of FSLID is about **11.25 times** and **3.16 times** faster. This proves the effectiveness of the proposed FSLID on real-world image datasets. From the Fig. 7, the visual results produced by FSLID are the best. To intuitively demonstrate the superiorities of our FSLID, the results of four baseline compared methods are visualized in Fig. 7. Distinctly, from this visualized example, four baseline methods are not apparent in removing complex real noise. This is mainly due to the lack of effective matching training strategies in these baseline methods, resulting in weak denoising effect. In all the methods, the visual results produced by FSLID are the best. This further proves the effectiveness of the FSLID and our few-shot two-stage training strategy.

E. Ablation Study of FSLID

We mainly analyze the denoising performance of the FSLID from the following aspects. The impact of the number of epochs in the first stage, number of training shots and the MSFR block on the denoising performance of the FSLID. All experiments are performed on synthetic and real noisy datasets.

Number of Epochs in the First Stage In our experiments, the BFL-SIR model in the first training stage only uses 200 epochs for training, while the DFI-SIR model in the second

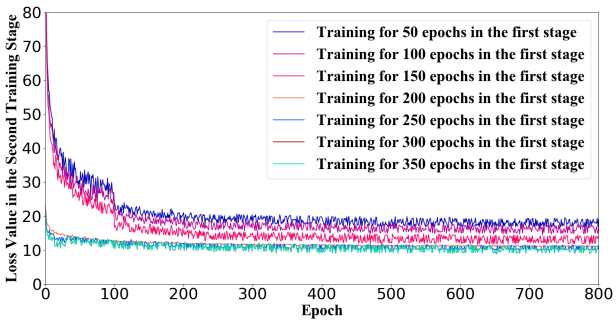

Fig. 8. Effects of the Epoch numbers in the first stage on the loss value of the FSID framework.

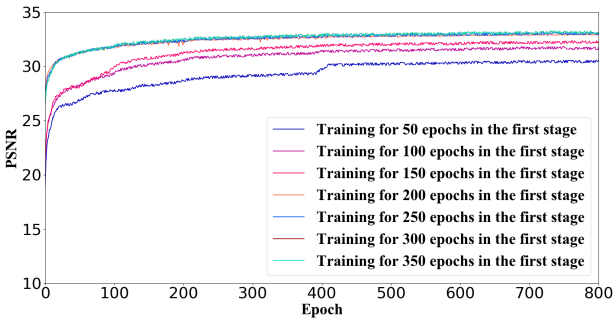

Fig. 9. Effects of the Epoch numbers in the first stage on the PSNR value of the FSLID.

training stage needs to extract guided features from the first stage for image reconstruction. Therefore, we focus on the impact of the number of iterations in the first stage on the denoising performance of the proposed FSLID. In order to ensure the fairness and objectivity of the ablation analysis experiment, we performed ablation experiments on the number of epochs in the first stage of training, and maintained 800 epochs in the second stage. As shown in Fig. 8 and Fig. 9, we found that when the first stage is kept within 200 epochs, as the number of epochs in the first stage increases, the denoising performance of the FSLID can be improved markedly. But after more than 200 epochs, as the number of epochs in the first stage increases, the denoising performance of FSLID is hardly improved. Therefore, in this work, 200 epochs are set in the first training stage of all the subsequent experiments.

Number of Training Shots In Tables VIII, with the increasing number of traning shots, the performances produced by our method first largely increase and then stabilizes. This proves that the performances will be saturated when using much more training shots. This is probably because in the second training phase, the induced knowledge learned from the first training phase is sufficient enough, much more training samples may not supplement the representations. Hence, the performance promotion is limited when using too many training shots. This further implies the effectiveness of the induced knowledge from the first training stage within our framework.

TABLE VIII. Ablation study of the number of training shots. PSNR values are from CBSD68 dataset.

Noisy level	K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80
l_n = 15	30.89	32.02	32.13	32.21	32.36	32.57	32.61	32.65
l_n = 25	27.85	29.04	29.11	29.23	29.42	29.63	29.68	29.70
l_n = 50	25.33	26.57	26.76	27.07	27.34	27.87	27.92	27.99

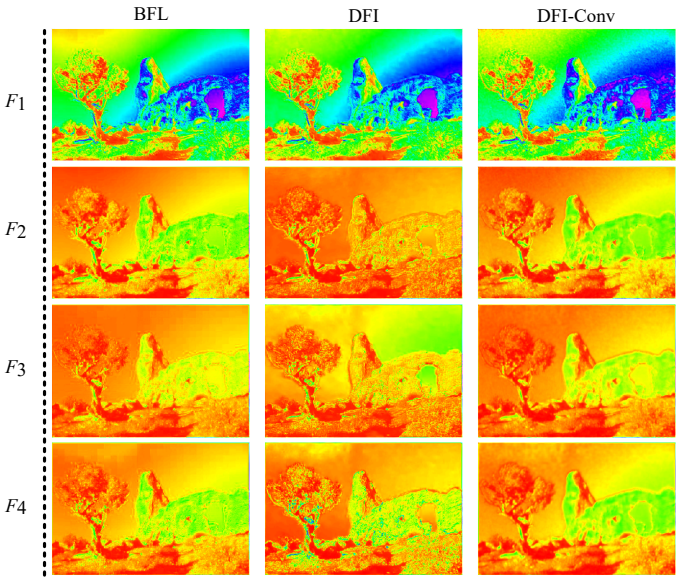

Fig. 10. Visualization effects of BFL for frozen weights, DFI and DFI-Conv in the FSLID on CBSD68 dataset ($l_n = 15$).

Ablation study of MSFR Block Since the structure of the DFI is mainly composed of MSFR blocks, in order to ensure the fairness and objectivity of the MSFR block ablation experiment, we replace the MSFR block in the FSLID with a regular convolutional layer with a kernel size of 3×3 to form the compared DFI-Conv structure. The parameters of both the DFI and SIR use the default parameter settings of the FSLID. We chose the $\mathcal{D}_{novel}^{S,K=20}$ dataset for training.

As can be seen from Table IX, compared with the FSLID without the MSFR block (*i.e.*, DFI-Conv), the MSFR block increases the PSNR and SSIM on all test datasets with $l_n = 15$ by 0.57 dB and 0.012 on average, and reduces LPIPS by 0.0076. This shows that the MSFR block is very effective to improve the denoising performance of the FSLID. Obviously, from the Fig. 10, compared with the DFI with multi-scale feature learned from MSFR blocks, the denoising prior features induced by DFI-Conv only generate coarse feature information, while DFI with MSFR blocks can produce finer features with abundant details, including rich texture feature information. This also shows that the DFI with MSFR block has a significant improvement for the proposed on FSLID.

F. Discussion of Limitations

Since $\mathcal{D}_{novel}^{K}$ is randomly selected from $\mathcal{D}_{novel}$, the data distribution in $\mathcal{D}_{novel}^{K}$ obeys the data distribution of $\mathcal{D}_{novel}$, *i.e.*, the data in $\mathcal{D}_{novel}^{K}$ is relatively single, which may affect the performance of FSLID. As shown in the table V, the model

TABLE IX. The quantitative results (PSNR ↑ /SSIM ↑ /LPIPS ↓) of DFI and DFI-Conv methods on the synthetic and real noisy datasets using the $\mathcal{D}_{novel}^{S,K=20}$ training dataset.

Test Datasets	DFI-Conv			DFI		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Set12	29.93	0.756	0.082	30.24	0.763	0.079
CBSD68	31.14	0.769	0.055	32.01	0.783	0.037
KODAK	31.63	0.772	0.041	32.16	0.786	0.039
SIDD	36.72	0.826	0.028	38.35	0.902	0.011

obtained using $\mathcal{D}_{novel}^{K}$ training has obvious differences in PSNR values on the test sets CBSD68, KODAK and Set12 with the same noise level. This also shows that when the randomly selected $\mathcal{D}_{novel}^{K}$ is close to the test set data distribution, the performance of FSLID may improve. On the contrary, when the extracted X has a large deviation from the data distribution of the test set, the performance of FSLID may decrease.

VI. Conclusion

This work attempts to propose a novel two-stage Few-Shot Learning for Image Denoising (FSLID). In the first stage, we introduce BFL and SIR for joint unsupervised training on $\mathcal{D}_{base}$. Such unsupervised training not only allows BFL to learn feature expressions efficiently, but also provides the induced and constraint information in the subsequent learning stage. In the second stage, we propose the DFI. DFI induces the information learned from the InPFL pre-trained in the first stage to generate the denoising prior features for the frozen SIR to adaptively reconstruct the denoised images. Additionally, a highly refined, low-channel-count, recursive Multi-Scale Feature Recursive (MSFR) is proposed to modularly formulate an efficient Denoising Feature Inducer (DFI) to capture a broader range of contextual feature information for a limited number of feature channels. Therefore, the proposed MSFR can reduce parameters and computation complexity significantly. Extensive experiments show that the proposed FSLID outperforms the baseline methods by a large margin, especially in using a scarce number of paired images. We hope this pioneering efficient FSLID scheme can encourage further exploration of the field of few-sample image denoising.

VII. Acknowledgment

This work was supported in part by the NSFC fund 62176077 and 62206073, in part by the Shenzhen Key Technical Project under Grant 2022N001, 2020N046, and 2022N063, in part by the Shenzhen Fundamental Research Fund under Grant JCYJ20210324132210025, in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), in part by the Guangdong Shenzhen joint Youth Fund under Grant 2021A151511074, and in part by the Natural Science Foundation of Guangdong Province under Grant 2023A1515010893.

References

[1] H. Yue, J. Liu, J. Yang, X. Sun, T. Q. Nguyen, and F. Wu, "Ienet: Internal and external patch matching convnet for web image guided denoising," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 3928–3942, 2020.

[2] B. Jiang, Y. Lu, J. Wang, G. Lu, and D. Zhang, "Deep image denoising with adaptive priors," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[3] A. Lahiri, S. Bairagya, S. Bera, S. Haldar, and P. K. Biswas, "Lightweight modules for efficient deep learning based image restoration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 1395–1410, 2021.

[4] W. Jiang, K. Huang, J. Geng, and X. Deng, "Multi-scale metric learning for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 1091–1102, 2021.

[5] J. Zhang, X. Zhang, and Z. Wang, "Task encoding with distribution calibration for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[6] S. Shao, L. Xing, R. Xu, W. Liu, Y. Wang, and B. Liu, "Mdfm: Multi-decision fusing model for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[7] L. Zhang, F. Zhou, W. Wei, and Y. Zhang, "Meta-generating deep attentive metric for few-shot classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[8] C. Zhang, C. Li, and J. Cheng, "Few-shot visual classification using image pairs with binary transformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 2867–2871, 2020.

[9] Y. Xiao, Y. Jin, and K. Hao, "Adaptive prototypical networks with label words and joint representation learning for few-shot relation classification," *IEEE transactions on neural networks and learning systems*, vol. PP, 2021.

[10] X. Lin, Z. Li, P. Zhang, L. Liu, C. Zhou, B. Wang, and Z. Tian, "Structure-aware prototypical neural process for few-shot graph classification." *IEEE transactions on neural networks and learning systems*, vol. PP, 2022.

[11] B. Yang, M. Lin, Y. Zhang, B. Liu, X. Liang, R. Ji, and Q. Ye, "Dynamic support network for few-shot class incremental learning." *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2022.

[12] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "Nbnet: Noise basis learning for image denoising with subspace projection," *ArXiv*, vol. abs/2012.15028, 2020.

[13] B. Jiang, Y. Lu, G. Lu, and D. Zhang, "Real noise image adjustment networks for saliency-aware stylistic color retouch," *Knowl. Based Syst.*, vol. 242, p. 108317, 2022.

[14] Z. Yue, H. Yong, Q. Zhao, L. Zhang, and D. Meng, "Variational denoising network: Toward blind noise modeling and removal," in *NeurIPS*, 2019.

[15] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, pp. 3142–3155, 2017.

[16] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, pp. 4608–4622, 2018.

[17] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. R. Fu, "Residual dense network for image super-resolution," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, 2018.

[18] S. Anwar and N. Barnes, "Real image denoising with feature attention," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3155–3164, 2019.

[19] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1712–1722, 2019.

[20] M. Chang, Q. Li, H. Feng, and Z. Xu, "Spatial-adaptive network for single image denoising," *ArXiv*, vol. abs/2001.10291, 2020.

[21] B. Jiang, J. Li, H. Li, R. Li, D. Zhang, and G. Lu, "Enhanced frequency fusion network with dynamic hash attention for image denoising," *Information Fusion*, vol. 92, pp. 420–434, 2023.

[22] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 240–252, 2022.

[23] C. Xu, C. Liu, L. Zhang, C. Wang, J. Li, F. Huang, X. Xue, and Y. Fu, "Learning dynamic alignment via meta-filter for few-shot learning," in *CVPR*, 2021.

[24] Z. Chen, J. Ge, H. Zhan, S. Huang, and D. Wang, "Pareto self-supervised training for few-shot learning," *ArXiv*, vol. abs/2104.07841, 2021.

[25] H. Zhang, H. Li, and P. Koniusz, "Multi-level second-order few-shot learning," *IEEE Transactions on Multimedia*, vol. abs/2201.05916, 2022.

[26] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 594–611, 2006.
[27] G. R. Koch, "Siamese neural networks for one-shot image recognition," 2015.
[28] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei, "Label efficient learning of transferable representations acrosss domains and tasks," in *NIPS*, 2017.
[29] L. Casas, A. Klimmek, G. Carneiro, N. Navab, and V. Belagiannis, "Few-shot meta-denoising," *arXiv preprint arXiv:1908.00111*, 2019.
[30] B. Jiang, J. Wang, Y. Lu, G. Lu, and D. Zhang, "Multilevel noise contrastive network for few-shot image denoising," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.
[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015.
[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.
[34] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1122–1131, 2017.
[35] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018.
[36] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 898–916, 2011.
[37] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," *arXiv preprint arXiv:1803.04189*, 2018.
[38] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2129–2137.
[39] C. Niu and G. Wang, "Noise2sim–similarity-based self-learning for image denoising," *arXiv e-prints*, pp. arXiv–2011, 2020.
[40] Z. Wang, A. Bovik, H. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.
[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.
[42] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," *arXiv preprint arXiv:2003.06957*, 2020.
[43] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "Hinet: Half instance normalization network for image restoration," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 182–192, 2021.
[44] S. W. Zamir, A. Arora, S. H. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," *ArXiv*, vol. abs/2003.06792, 2020.